

RALLYE SCIENCES 2023

Sciences Numériques

Classe : **Sciences numériques**

Établissement :

Commune :

Département :

Etiquette à découper et à coller sur l'affiche

La stéganographie

Matériel nécessaire :

- ordinateur avec accès internet et les fichiers liés à l'énoncé ;
- une affiche A3 ;
- des crayons de couleurs ou feutres éventuellement ;
- une imprimante éventuellement.

Production attendue :

Une affiche A3 avec :

- au recto, les éléments de votre recherche de la partie 1 ;
- au verso, les réponses de la partie 2.

Sur cette affiche sera obligatoirement indiquée en haut à droite la mention « Rallye Sciences 2023 – Partie Sciences Numériques » ainsi que le nom de l'établissement, la ville et la classe.

Remarque : mise à part la question 1 de la partie 2 qui demande de reconnaître les couleurs, toutes les autres questions peuvent être traitées par tous les élèves.

Partie 1

Voici une image représentant un tableau de Pollock :

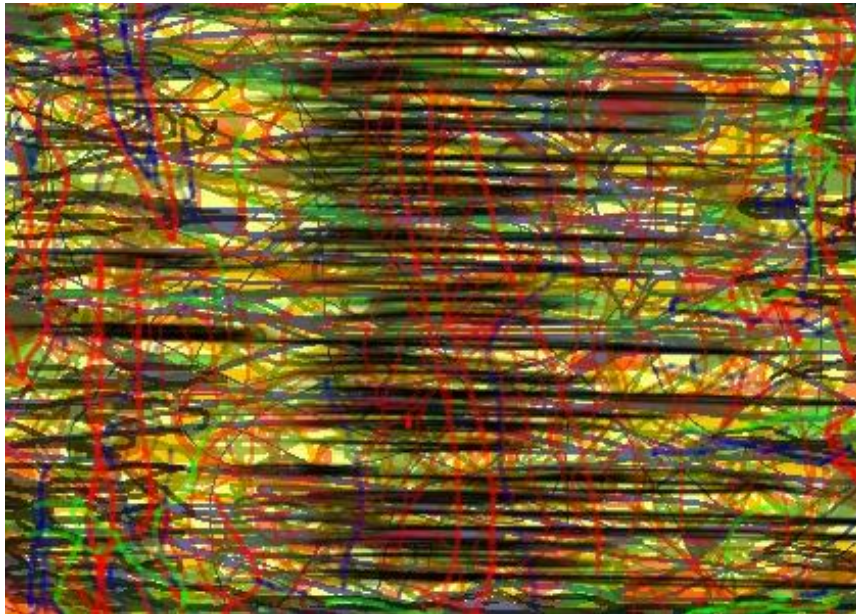
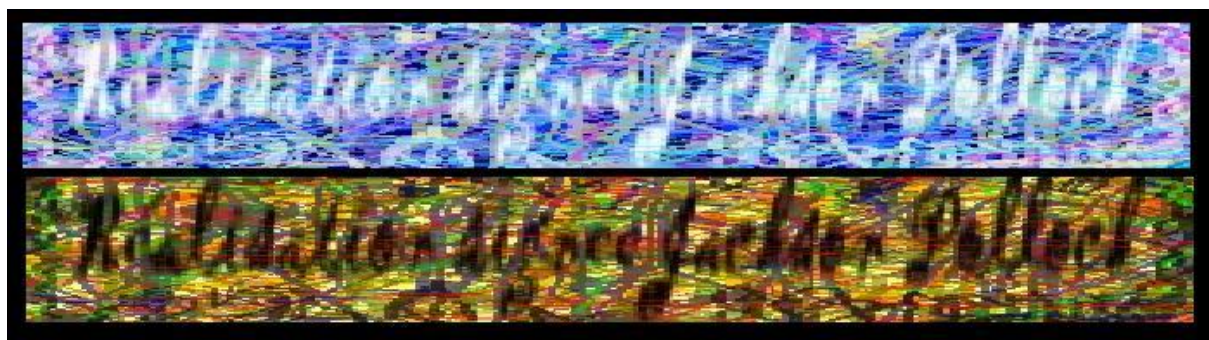


Figure 1

Nos yeux nous montrent-ils vraiment la réalité ?

Voilà ce qui se cache dans cette image...



Réalisation signée Jackson Pollock

Figure 2

Ce procédé, rendu possible grâce à une technique de codage qui modifie le code RVB de l'image, s'appelle la stéganographie.

Rechercher des informations concernant :

- la stéganographie ;
- le code RVB (ou RGB en anglais) de 0 à 255 ;
- le code ASCII.

Rédiger et illustrer avec esthétisme une synthèse de vos recherches sur le recto d'une affiche de taille A3.

Voici quelques exemples de questionnement pour guider votre recherche :

- Quel est l'intérêt de la stéganographie ? Est-ce une technique récente ? Quelles méthodes peuvent être utilisées dans ce domaine ?
- A quoi sert le code RVB ? Existe-t-il d'autres façons de coder en RVB ? Des valeurs supérieures à 255 peuvent-elles être utilisées ?
- Quelle est l'utilité du code ASCII ?

Ne pas hésiter à se répartir le travail avant une mise en commun.

Partie 2

Les réponses aux questions de cette partie seront rédigées au verso de l'affiche A3 utilisée dans la partie précédente.

Il s'agit dans cette partie d'étudier plusieurs méthodes pour cacher des informations dans des images afin qu'un expéditeur puisse les transmettre par voie numérique de manière secrète à un destinataire.

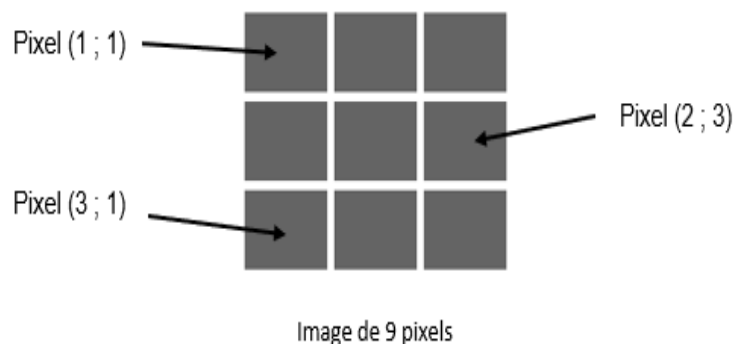
On suppose ainsi que le destinataire et l'expéditeur du message caché disposent de l'image initiale. L'expéditeur cache le message dans l'image initiale, envoie l'image modifiée au destinataire, qui peut ainsi retrouver le message caché en travaillant sur les différences entre les deux images.

Les images de la partie 1 comportent un grand nombre de pixels. Pour simplifier, nous allons nous intéresser à des images beaucoup plus simples comportant 9 pixels seulement. Mais le principe reste le même !

Le but de cette partie est ainsi de travailler sur des méthodes efficaces permettant de cacher des caractères (des lettres par exemple) dans une image comportant 9 pixels (3 pixels par 3 pixels).

Convention pour la suite :

La position d'un pixel dans une image sera indiquée par un couple (n ; p) où n est le numéro de la ligne où se trouve le pixel, en partant du haut, et p celui de la colonne en partant de la gauche. Par exemple, le pixel à la position (1 ; 1) est le « premier » pixel qui se situe en haut à gauche de l'image.



Méthode 1.

1. Ouvrir le fichier *partie2question1.html*.

En comparant attentivement l'image modifiée à gauche et l'image initiale à droite, ainsi que les niveaux de rouge, de vert et de bleu dans le codage RVB de chaque pixel, indiquer dans quel pixel se trouve le caractère caché « B ».

2. On a utilisé l'algorithme ci-dessous, écrit en langage naturel, pour cacher un caractère dans un pixel (l'algorithme peut être généralisé facilement pour cacher plusieurs caractères et donc un message !).

Numéro de lignes	Algorithme 1
1	Enregistrer l'image dans un nouveau fichier
2	Choisir un pixel dans lequel on veut cacher le caractère
3	$\text{nivBleu} \leftarrow \text{Niveau de bleu du pixel choisi}$
4	$\text{Nb} \leftarrow \text{Nombre décimal représentant dans le code ASCII le caractère à cacher}$
5	$\text{nivBleu} \leftarrow \text{nivBleu} + \text{Nb}$
6	Mettre le niveau de bleu du pixel choisi à la valeur nivBleu

À l'aide de cet algorithme, vous souhaitez envoyer le caractère « A » (majuscule) caché dans le pixel (3,3) à partir de la même image initiale.

- a) Ouvrir le fichier *partie2questions2et3.html*.

- b) Donner le codage RVB du pixel à la position (3 ; 3) de l'image initiale (image à droite).
- c) Déterminer le codage RVB du pixel à la position (3 ; 3) de l'image modifiée, c'est-à-dire avec le caractère « A » (majuscule) caché dans le pixel (3 ; 3).

3.

- a) Avec l'algorithme 1 proposé, peut-on cacher dans l'image initiale du fichier *partie2questions2et3.html* le caractère « z » (lettre minuscule dont le code ASCII est 122) dans le pixel (3 ; 1) ? dans le pixel (3 ; 2) ? Pourquoi ?
- b) Expliquer alors quel problème cela peut engendrer lorsqu'on veut cacher un message dans une image.
- c) Pour résoudre ce problème, nous pouvons modifier l'algorithme 1 précédent de la façon suivante :

Numéro de lignes	Algorithme 2
1	Enregistrer l'image dans un nouveau fichier
2	Choisir un pixel dans lequel on veut cacher le caractère
3	NivBleu \leftarrow Niveau de bleu du pixel choisi
4	Nb \leftarrow Nombre décimal représentant dans le code ASCII le caractère à cacher
5	Si NivBleu + Nb \leq 255
6	NivBleu \leftarrow NivBleu + Nb
7	Sinon
8	NivBleu \leftarrow NivBleu + Nb - 256
9	Mettre le niveau de bleu du pixel choisi à la valeur NivBleu

On souhaite cacher la lettre « C » majuscule dans le pixel (1 ; 3) et la lettre « z » minuscule dans le pixel (2 ; 3) dans l'image initiale du fichier *partie2questions2et3.html*.

Donner les codages RVB des deux pixels contenant les deux lettres cachées, obtenus en appliquant l'algorithme 2.

- d) Montrer qu'avec l'algorithme 2, le niveau de bleu d'un pixel dans lequel un caractère est caché sera toujours compris entre 0 et 255.
- e) Proposer un algorithme permettant de retrouver un caractère caché par l'algorithme 2 dans un pixel choisi.

Avec cet algorithme simple, un pirate informatique peut, s'il intercepte le message contenant l'image modifiée et s'il connaît l'image initiale, obtenir très facilement les pixels modifiés et donc le message caché.

Dans les questions suivantes, on va ainsi imaginer des méthodes de chiffrement plus complexes qui cachent des caractères dans l'image, tout en modifiant leur codage.

Dans la question 4, on va utiliser la division euclidienne pour cacher un caractère dans l'image initiale.

On utilisera ici le fichier *partie2question4.html*.

Il est possible avec ce fichier de cliquer sur chaque pixel, d'entrer le code RVB voulu puis de cliquer sur « générer le tableau » pour obtenir un tableau contenant tous les codes RVB de l'image constituée de 9 pixels.

Méthode 2.

4. Par exemple, pour la lettre majuscule « A », on effectue la division euclidienne de 65 par 3. Cela donne : $65 = 3 \times 21 + 2$. On ajoute (ou soustrait si le niveau dépasse 255) le nombre 21 au niveau de rouge du pixel choisi, 21 à son niveau de vert et 23 à son niveau de bleu.

- a) On souhaite cacher la lettre « A » majuscule dans le pixel (1,2) dans l'image initiale du fichier *partie2question4.html*.

Donner, en appliquant cette méthode, le code RVB de ce pixel sans la lettre cachée puis le code de ce pixel avec la lettre cachée.

- b) Même question pour la lettre majuscule « B » dans le pixel (1,3) puis la lettre majuscule « C » dans le pixel (2,3).
- c) Que penser de cette méthode de chiffrement ? Argumenter (on pourra par exemple comparer cette méthode avec l'algorithme 2).

Méthode 3 : la vôtre !

5.

- a) Proposer une autre méthode efficace pour cacher un caractère dans un pixel codé en RVB.

- b) Cryptage :

Donner en langage naturel l'algorithme associé à votre méthode qui permet de cacher un caractère dans un pixel codé en RVB.

- c) Décryptage :

Donner en langage naturel l'algorithme associé à votre méthode qui permet de retrouver quel est le caractère caché dans l'image modifiée.

- d) On souhaite cacher le mot « Rallye » dans l'image de Pollock de l'introduction.

Pour simplifier, on considère un détail de cette image : il s'agit d'une forme carrée de 100 pixels, reproduite dans le fichier *partie2question5.html* (on pourra visionner le fichier *partie2question5_zoom_pollock.mp4*).

Les six lettres du mot « Rallye » (attention à la majuscule !) seront cachées dans les pixels en diagonale (du pixel (1 ; 1) au pixel (6 ; 6)) de l'image du fichier *partie2question5.html*.

Indiquer ainsi, sur le verso de l'affiche A3, les codes RVB des six pixels modifiés en appliquant votre méthode. Nous devons être capables de décoder le mot en suivant vos explications de la question 5.c !